

Creating and Modifying an Isofill Graphics Method

Â

[Contents](#) [Previous](#) [Next](#)

Goal: Guide you through creating and setting isofill graphics method attributes.

Before running the tutorial below, type "*python*" or "*cdat*" at the command line.Â You will see the python prompt appear (i.e., ">>>"). You can now enter the command lines below.

You can [view](#)Â or [download](#)Â the full source code. To run the source code at the command line, type: "*python isofill_file.py*".

```
# Import the modules needed for the tutorial
# cdms - Climate Data Management system accesses gridded data.
# vcs - Visualization and control System 1D and 2D plotting routines.
# cdutil - Climate utilitizes that contains miscellaneous routines for
#           manipulating variables.
# time - This module provides various functions to manipulate time values.
# os - Operation System routines for Mac, DOS, NT, or Posix depending on
#       the system you're on.
# sys - This module provides access to some objects used or maintained by
#       the interpreter and to functions that interact strongly with the interpreter.
import vcs, cdms, cdutil, time, os, sys

# Open data file:
filepath = os.path.join(sys.prefix, 'sample_data/clt.nc')
cdmsfile = cdms.open( filepath )

# Extract a 3 dimensional data set and get a subset of the time dimension
data = cdmsfile('clt', longitude=(-180, 180), latitude = (-90., 90.))

# Initial VCS:
v = vcs.init()

# Show the list of persistent isofill graphics methods.
v.show('isofill')

*****Isofill Names List*****
( 1):          ASD          ASD_map        P_and_height
( 4):          default      polar          quick
( 7):          robinson
*****End Isofill Names List*****
```

Get a isofill graphics method object and plot:

```
# Assign the variable "cf_asd" to the persistent 'ASD' isofill graphics methods.
cf_asd = v.getisofill( 'ASD' )

# Plot the data using the above isofill graphics method.
v.plot( data, cf_asd )
```

List the 'ASD' isofill graphics method attributes by issuing the following command:

```
# List the 'ASD' isofill graphics methods attributes.
cf_asd.list()
```

```
-----Isofill (Gfi) member (attribute) listings -----
Canvas Mode = 1
graphics method = Gfi
name = ASD
projection = linear
xticlabels1 = *
xticlabels2 = *
xmtics1 = *
xmtics2 = *
yticlabels1 = *
yticlabels2 = *
ymtics1 = *
ymtics2 = *
datawc_x1 = 1.00000002004e+20
datawc_y1 = 1.00000002004e+20
datawc_x2 = 1.00000002004e+20
datawc_y2 = 1.00000002004e+20
datawc_timeunits = days since 2000
datawc_calendar = 135441
xaxisconvert = linear
yaxisconvert = linear
missing = 1.00000002004e+20
ext_1 = n
ext_2 = n
fillareastyle = solid
fillareaindices = None
fillareacolors = [241]
levels = ([1.0000000200408773e+20, 1.0000000200408773e+20],)
legend = None
```

Change 'ASD' isofill graphics methods attributes by entering the appropriate command lines:

```
# change the isofill levels and the color indices
cf_asd.levels = ([0,20],[20,40],[50,60],[60,70],[70,80],[90,100])
cf_asd.fillareacolors=[ 22,44,66,88,110,132])
```

```
# change the isofill levels and the color indices
cf_asd.levels = ([0,20,25,30,35,40],[70,75,80], [90,100])
cf_asd.fillareacolors=[ 22,44,66,88,110,132,152,174])
```

```
# Change the levels back to the default settings.
cf_asd.levels = ([1e20], )
```

```
# Change the colormap from rainbow to green to magenta.
v.setcolormap('grn_to_magenta')
```

Create hatches and pattern fill areas.

```
# Change the colormap back to rainbow.
v.setcolormap('rainbow')

# Specify levels
cf_asd.levels = ([0,20],[20,40],[50,60],[60,70],[70,80],[90,100])

cf_asd.fillareastyle='pattern' # Change fill style to pattern
```

```
cf_asd.fillareastyle='hatch'# Change fill style to hatch
cf_asd.fillareaindices=([1,3,5,6,9,20])# Hatch patterns
cf_asd.fillareacolors=([22, 44, 88, 122, 144, 188]) # Change color

# Create a persistent isofill graphics methods from an existing isofill graphics method.
cf_new = v.createisofill( 'new', 'ASD' ) # create new from ASD
cf_new2 = v.createisofill( 'new2','quick' )# create new2 from quick
cf_new.list()                         # list its attributes
v.show('isofill')                     # show isofill list with new and new2
v.removeobject( cf_new )              # remove new from isofill list
v.show('isofill')                     # show isofill list without new
v.removeobject( cd_new2 )             # remove new2
v.show('isofill')                     # show isofill list
```

Â

[Contents](#) [Previous](#) [Next](#)